МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение

высшего образования

«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра «Измерительно-вычислительные комплексы»

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | |  | |  |  |
| |  |  | | --- | --- | | ***Подп. и дата*** |  | | ***Инв. № дубл.*** |  | | ***Взам. инв. №*** |  | | ***Подп. и дата*** |  | | ***Инв. № подл*** |  | |  | | ПОЯСНИТЕЛЬНАЯ ЗАПИСКА  на лабораторную работу №9  по дисциплине «Алгоритмы и структуры данных»  Тема «Игра крестики-нолики» | | | | | | |
|  | | |  | |  | | | |
|  | | Исполнитель  студент гр. ИСТбд-21 Мишин А.В.  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_ 2024 г. | | | |
|  | | 2024 | | | | | | |

**1. Введение**

Разрабатываемое приложение представляет собой игру "Крестики-нолики", созданную с использованием языка программирования Python и библиотеки Tkinter для графического интерфейса. Целью приложения является реализация классической игры для двух игроков (или одного игрока против компьютера) с возможностью отображения состояния игры, определения победителя, а также подсчета времени игры.

**2. Основания для разработки:**

Задание для лабораторной работы по учебному плану дисциплины «Алгоритмы и структуры данных» на направление "Информационные системы и технологии".

**3.Постановка задачи**

Цель разработки заключается в создании игрового приложения, которое позволяет пользователям играть в "Крестики-нолики". Игра должна иметь два режима: игра против другого игрока и игра против компьютера. Также приложение должно отображать текущий ход, определять победителя, учитывать время игры и предоставлять возможность перезапуска игры.

**4. Структура программы**

**4.1 Классы:**

Программа состоит из одного основного класса, который управляет логикой игры и интерфейсом.

* Game: основной класс программы, реализующий логику игры и взаимодействие с интерфейсом.

**4.2 Основные методы класса:**

* create\_widgets(): создание и размещение всех элементов интерфейса.
* start\_game(): запуск игры в выбранном режиме.
* button\_click(index): обработка нажатия на клетку игрового поля.
* check\_winner(): проверка победителя.
* computer\_turn(): выполнение хода компьютера (для режима "1 игрок").
* switch\_turn(): переключение хода между игроками.
* reset\_game(): перезапуск игры.
* update\_timer(): обновление таймера игры.

**4.3 Функциональное назначение:**

* Игра должна поддерживать два режима: игра против игрока и игра против компьютера.
* После каждого хода игрока или компьютера игра должна проверять состояние поля и определять победителя.
* Время игры должно отображаться на экране.
* Игра должна автоматически переключать ход между игроками или делать ход компьютером.
* После завершения игры пользователь должен увидеть сообщение о победителе или ничьей.
* В игре должна быть кнопка для перезапуска и кнопка для выхода в главное меню.

**5. Проектная часть**

**5.1 Математические методы**

Программа использует операции с датами и временем для отслеживания времени игры. Алгоритм Minimax, используемый для игры против компьютера, опирается на рекурсивное дерево поиска с оценкой каждого состояния игры.

**5.2 Архитектура и алгоритмы**

**5.2.1 Архитектура**

Программа состоит из нескольких ключевых частей:

* Главное окно приложения — окно с элементами управления.
* Игровое поле — кнопки, представляющие клетки поля 3x3.
* Панель управления — кнопки для перезапуска игры, выхода в главное меню.
* Таймер — отображение времени игры.

**5.2.2 Алгоритмы**

* Алгоритм игры: Логика игры определяется с помощью проверки состояния поля после каждого хода, поиска победных комбинаций, переключения игроков.
* Алгоритм Minimax (для режима игры против компьютера): используется для вычисления наилучшего хода для компьютера на основе оценки состояния поля.

**6. Стадии и этапы разработки**

1. Определение требований и задач.
2. Проектирование структуры классов и интерфейса.
3. Реализация интерфейса с использованием библиотеки Tkinter.
4. Реализация логики игры.
5. Тестирование функциональности игры.
6. Оптимизация и улучшение интерфейса.

**7. Тестирование**

**7.1 Цель тестирования**

Цель тестирования заключается в проверке корректности работы всех функций игры. В ходе тестирования проверяются:

**7.2 Методика тестирования**

Тестирование проводилось методом "черного ящика", где проверялись функциональные возможности приложения при различных сценариях использования.

**7.3 Тестирование**

1. Корректность работы алгоритмов игры.
2. Отображение состояния игры.
3. Обработка всех возможных состояний поля (победа, ничья, переключение хода).
4. Работа интерфейса (кнопки, отображение времени).
5. Стабильность работы программы при разных сценариях использования.

**7.4 Результаты тестирования**

Тестирование показало, что программа корректно обрабатывает все основные функции: переключение ходов, проверка победителя, отображение времени игры, а также работу интерфейса.

**8. Заключение**

Разработанная программа полностью выполняет поставленные задачи и обеспечивает удобный и функциональный интерфейс для игры в "Крестики-нолики". Реализован режим игры против другого игрока и против компьютера с логикой, основанной на алгоритме Minimax.

**9. Источники, использованные при разработке**

1. Лутц М. Изучаем Python. – СПб.: Символ-Плюс, 2011.

2. "Документация Tkinter" [Электронный ресурс] // Официальная документация Python: https://docs.python.org/3/library/tkinter.html – Режим доступа: свободный.

3. Саммерфилд М. Python на практике. – М.: ДМК Пресс, 2014.

4. ГОСТ 19.701-90 «Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения». – М.: Издательство стандартов, 1990.

5. Бизли Д. Python. Подробный справочник. – СПб.: Символ-Плюс, 2010.